RTU Liepājas akadēmija

Dabas un inženierzinātņu centrs

Endijs Dārznieks

Profesionālā bakalaura studiju programmas   
„Informācijas tehnoloģija”   
2.kursa students

**Code Moodle**

Studiju darbs

Zinātniskie vadītāji –   
Lekt. Dzintars Tomsons

Dr. sc. comp., Prof. Anita Jansone

Liepāja 2025

**Anotācija**

**Darba autors:** Endijs Dārznieks

**Darba tēma:** Code Moodle

**Darba veids:** Studiju darbs

**Studiju programma:** Informācijas tehnoloģija

**Darba zinātniskie vadītāji:** Lekt. Dzintars Tomsons, Dr. sc. comp., Prof. Anita Jansone

**Darba apjoms:** aptuveni 20 lpp.

**Atslēgas vārdi:** datubāzu pārvaldības sistēmas, datubāzu arhitektūra, programmatūras inženierija, datorsistēmu admninistrēšana, programmatūras projekta pārvaldība

**Pētījuma mērķis:** Noteikt piemērotāko datubāzu pārvaldības sistēmu oriģinālas satura pāvaldības sistēmas izstrādei.

**Darba saturs:** Darbs sastāv no 2. daļām.

Darba I daļa ir autora teorētiskais pētījums par piemērotāko datubāzu pārvaldības sistēmu oriģinālas satura pāvaldības sistēmas izstrādei. Daļa satāv no ievada, divām nodaļām, secinājumiem un izmantotās literatūras saraksta.

1.nodaļa satur aprakstu par datubāzu uzbūvi un datubāzu pārvaldības sistēmu attīstības vēsturi.

2.nodaļā iekļauts iekļauts šobrīd aktuālāko datubāzes pārvaldības sistēmu apskats un to salīdzinošās analīze – novērtēšanas kritēriju pamatojums, programmatūras novērtējums pēc izvēlētajiem kritērijiem un analīzes rezultātu apraksts.

Darba II daļa ir komandas projekta “Code Moodle” izstrādātās programmatūras apraksts. 1.nodaļā raksturot programmatūras izstrādes projektu.

2.nodaļā dotas projekta nodevumu apraksts.

3.nodaļa raksturo komandas dalībnieku un autora ieguldījumu komandas projektaīstenošanā.

**Pētījuma metodes:** Literatūras studijas, datubāzu pārvaldības sistēmu salīdzinošā analīze, izstrādātās programmatūras sistēmas aprobācija.

**Darba rezultāti:** Datubāzu pārvaldības sistēmu salīdzinošā analīze, programmatūras projekta apraksts, programmatūras projekta dokumentācija apraksts.

**Darba izmantojamība:** Projektā izstrādāto platformu paredzēts izmantot sadarbības veicināšanai starp studentiem, skolotājiem un akadēmisko personālu. Darba I daļu var izmantot studenti un citi interesenti pamatzināšanu ieguvei par datubāzu uzbūvi. Savukārt raksturotos datubāzu pārvaldības sistēmu novērtēšanas kritēriju var izmantot līdzīgu projektu izstrādē programmatūras rīku izvēles noteikšanai.

**Annotation**

**Author of the paper:** Endijs Dārznieks

**Theme of the paper:** Code Moodle

**Type of the paper:** Annual project

**Study program:** Information techonolgy

**Consultants of the paper:** Lekt. Dzintars Tomsons, Dr. sc. comp., Prof. Anita Jansone

**Volume of the paper:** about 20 pages

**Key words:** database management systems, database architecture, software engineering, administration of computer systems, software project management

**Aim of the research**: To determine the most suitable database management system for the development of an original content management system.

**Content of the paper:** The paper consists of two parts.

Part I is the author's theoretical research on the most appropriate database management system for developing an original content management system. This part includes an introduction, two chapters, conclusions, and a list of references.

Chapter 1 provides a description of the structure of databases and the historical development of database management systems.

Chapter 2 includes an overview of the most relevant current database management systems and their comparative analysis — justification of evaluation criteria, software evaluation according to the selected criteria, and a description of the analysis results.

Part II describes the software developed within the team project "Code Moodle".

Chapter 1 characterizes the software development project.

Chapter 2 provides a description of the project deliverables.

Chapter 3 outlines the contributions of the team members and the author to the implementation of the team project.

**Methods of research:** Literature review, comparative analysis of database management systems, testing of the developed software system.

**Results of the research:** Comparative analysis of database management systems, description of the software project, and documentation of the software project.

**Applicability of the paper:** The platform developed in the project is intended to facilitate collaboration between students, teachers, and academic staff. Part I of the paper can be used by students and other interested parties to gain basic knowledge about database structure. The described evaluation criteria for database management systems can be used in similar projects to determine the appropriate software tools.

**Saturs**

[I daļa Datubāzes, to veidošanas pieejas un rīki 5](#_Toc199434618)

[Ievads 5](#_Toc199434619)

[1. Datubāzu pārvaldības sistēmas un to attīstības vēsture 6](#_Toc199434620)

[2. Datubāzu pārvaldības sistēmu salīdzinošā analīze 7](#_Toc199434621)

[2.1. Vērtēšanas kritēriji 7](#_Toc199434622)

[2.2. Datubāzu pārvaldības sistēmas 12](#_Toc199434623)

[2.2.1. Oracle Database 12](#_Toc199434624)

[2.2.2. MySQL 13](#_Toc199434625)

[2.2.3. Microsoft SQL Server 14](#_Toc199434626)

[2.2.4. PostgreSQL 15](#_Toc199434627)

[2.2.5. MongoDB 16](#_Toc199434628)

[2.2.6. IBM Db2 17](#_Toc199434629)

[2.2.7. SQLite 18](#_Toc199434630)

[2.3. Datubāzu pārvaldības sistēmu salīdzinošā analīze 19](#_Toc199434631)

[2.4. Analīzes rezultāti 20](#_Toc199434632)

[Secinājumi 21](#_Toc199434633)

[Jēdzienu skaidrojumi un saīsinājumi 22](#_Toc199434634)

[Izmantotā literatūra 24](#_Toc199434635)

[II daļa Programmatūras projekts, komandas veikums 26](#_Toc199434636)

[1. “Code Moodle” projekta apraksts 26](#_Toc199434637)

[1.1 Projekta mērķis un uzdevumi 26](#_Toc199434638)

[1.2 Projekta organizatoriskā struktūra 26](#_Toc199434639)

[1.3 Projekta nodevumi 27](#_Toc199434640)

[2. Projekta nodevumu apraksts 27](#_Toc199434641)

[3. Autora un pārējo komandas dalībnieku ieguldījums projekta tapšanā 28](#_Toc199434642)

# I daļa Datubāzes, to veidošanas pieejas un rīki

# Ievads

Šajā darbā raksturoti pētījuma rezultāti, kas iegūti projekta „Code Moodle” īstenošanas gaitā. Studiju projekta mērķis ir izstrādāt satura pārvaldības sistēmu studentu un skolotāju savstarpējai komunikācijai, uzdevumu, moduļu došanai un izpildei, informācijas kopskata apkopojumu savākšānai un attēlošanai utt. Projekta izstrādē piedalījās RTU Liepājas akadēmijas profesionālā bakalaura studiju programmas “Informācijas tehnoloģija” 2.kursa studenti Kārlis Lācītis un šī darba autors.

Studiju projekta ietvaros autora veiktā pētījuma mērķis:

Noteikt piemērotāko datubāzu pārvaldības sistēmu satura pāvaldības sistēmas “Code Moodle” izstrādei.

Darba mērķa sasniegšanai ir noteikti šādi darba uzdevumi:

1. Apkopot informāciju par datubāzu pārvaldības sistēmām un to lietojumiem satura pārvaldības sistēmu izstrādē.
2. Noteikt kritērijus datubāzu pārvaldības sistēmu novērtēšanai.
3. Noteikt un atlasīt satur pārvaldības sistēmu izstrādei piemērotākās datubāzu sistēmas.
4. Veikt atlasīto datubāzu pārvaldības sistēmu salīdzinošu analīzi un noteikt “Code Moodle” projektam piemērotāku datubāzu pārvaldības sistēmu.

Studiju projekta pētījumā izmantotas šādas pētījuma metodes:

1. literatūras un citu informācijas avotu studijas;
2. Programmatūras istrādes rīku salīdzinošā analīze;
3. Izstrādātās satura pārvaldības sistēmas aprobācija.

Darba pētījuma daļa sastāv no ievada, divām nodaļām, secinājumiem, skaidrojumu un literatūras saraksta. Darba 1.nodaļa satur aprakstu par datubāzu uzbūvi un datubāzu pārvaldības sistēmu attīstības vēsturi. 2.nodaļā iekļauts iekļauts šobrīd aktuālāko datubāzes pārvaldības sistēmu apskats un to salīdzinošās analīze – novērtēšanas kritēriju pamatojums, programmatūras novērtējums pēc izvēlētajiem kritērijiem un analīzes rezultātu apraksts.

# 1. Datubāzu pārvaldības sistēmas un to attīstības vēsture

Datubāzu pārvaldības sistēmas (DBPS) ir būtiska informācijas tehnoloģiju sastāvdaļa, kas ļauj efektīvi organizēt, glabāt un nolasīt datus. To attīstība ir cieši saistīta ar datoru tehnoloģiju progresu un mainīgajām uzņēmumu un organizāciju vajadzībām. DBPS attīstības vēsture aptver vairākus nozīmīgus posmus, sākot no hierarhiskām un tīkla datu struktūrām līdz mūsdienu mākoņdatošanas un autonomajām datubāzēm. (Team EMB, 2024) Piemēram, kā NoSQL tiek lēnām daudzkur adoptēts, lielie uzņēmumi to izmanto vienā vai citā veidā. (Perdue, 2024)

Sākotnējie datubāzu risinājumi parādījās 1960. gados ar tādām sistēmām kā Integrated Data Store (IDS), kuru izstrādāja Charles W. Bachman (Wikipedia, 2025). IDS bija pirmais, kas ieviesa tīkla modeli, ļaujot sarežģīti saistīt datu ierakstus (Sande, 2025). Ap to pašu laiku IBM radīja savu hierarhisko DBPS – Information Management System (IMS), tā kļuva par plaši izmantotu uzņēmumu vidē. (Quickbase, 2024) Šīs pirmās sistēmas nebija ļoti dinamiskas, tāpēc vēlāk bija nepieciešamība pēc jaunām sistēmām.

1970. gadā Edgar F. Codd publicēja relāciju datubāzu modeli, kurā dati tika organizēti tabulās. Šis modelis kļuva par pamatu modernajām relāciju DBPS, piemēram, Oracle, IBM DB2 un Microsoft SQL Server. IBM arī izstrādāja eksperimentālu sistēmu System R, kas demonstrēja šī modeļa efektivitāti. (Quickbase, 2024) Relāciju DBPS kļuva dominējošas 1980. un 1990. gados, jo tās piedāvāja standartizētu SQL vaicājumu valodu, dinamisku datu apstrādi un augstu veiktspēju.

Līdz ar interneta attīstību un datu apjoma pieaugumu 2000. gados, radās vajadzība pēc jauniem datubāzu risinājumiem, kas būtu piemēroti lielajiem un dažādajiem datiem. Šo izpildija NoSQL datubāzes, piemēram, MongoDB, Cassandra un CouchDB, šie risinājumi neievieš striktas tabulu struktūras, bet tomēr nodrošina horizontālu mērogojamību (Perdue, 2024). Tajā pašā laikā attīstījās arī "lielo datu" infrastruktūra, piemēram, Hadoop un Apache Spark, kas spēja paralēli apstrādāt milzīgus datu apjomus (CelerData, 2024) (Team EMB, 2024).

Mūsdienās DBPS klāsts ir vēl vairāk paplašinājies. Ir pieejami daudz un dažādi rīki: relāciju sistēmas (PostgreSQL, MySQL), NoSQL risinājumi (MongoDB), grafu datubāzes (Neo4j) un in-memory sistēmas (Redis). Vienlaikus notiek pāreja uz mākoņdatošanas platformām (DBaaS), piemēram, Amazon RDS, Google Cloud Spanner un Microsoft Azure SQL, kas nodrošina drošību un automātisku mērogojamību (Robb, 2023) (CelerData, 2024).

Attīstības tendencēs izceļas vairāki virzieni. Pirmkārt, pieaug interese par autonomām datubāzēm, kas spēj pašas veikt darbības bez cilvēka iejaukšanās, piemēram, izveidot un atomātiski atjaunot rezerves kopijas. Otrkārt, arvien vairāk DBPS tiek integrētas ar mākslīgā intelekta un mašīnmācīšanās risinājumiem, kas uzlabo datu analīzi, prognozēšanu, anomāliju atpazīšanu un sniedz atbalstu lēmumu pieņemšanai (Robb, 2023).

Tajā pašā laikā šīs inovācijas rada arī jaunus izaicinājumus, piemēram, nepieciešamība pēc labākas datu drošības, kvalitātes nodrošināšanas un saderības ar esošajām sistēmām ir aktuālāka nekā jebkad agrāk (Robb, 2023).

Kopumā DBPS attīstība atspoguļo informācijas tehnoloģiju kopējo virzību, kas ir no statiskas datu glabāšanas uz dinamisku, inteliģentu un mērogojamu datu pārvaldību.

# 2. Datubāzu pārvaldības sistēmu salīdzinošā analīze

## 2.1. Vērtēšanas kritēriji

1. Veiktspēja
   1. Veiktspēja ir būtiska, jo tā tieši ietekmē sistēmas efektivitāti un lietotāju mijiedarbību, kā arī apmierinātību ar sistēmu.
   2. Veiktspēja ir viens no galvenajiem DBPS izvērtēšanas kritērijiem, jo tā nosaka, cik ātri un efektīvi sistēma spēj apstrādāt datus (Taipalus, 2023) (Besa, 2024).
   3. Izmantojot standarta testus, piemēram, TPC-C, lai novērtētu sistēmas reakcijas laiku.
2. Mērogojamība
   1. Lai sistēma spētu pielāgoties pieaugošam datu apjomam un lietotāju skaitam.
   2. Mērogojamība nodrošina sistēmas ilgtermiņa efektivitāti un spēju apstrādāt lielus datu apjomus bez veiktspējas zuduma (Besa, 2024).
   3. Novērtējot sistēmas veiktspēju pie dažādiem datu apjomiem un lietotāju skaita pieauguma scenārijiem.
3. Uzticamība
   1. Uzticama sistēma nodrošina nepārtrauktu darbību un datu pieejamību.
   2. Uzticamība ir kritiska, lai nodrošinātu datu integritāti un sistēmas darbības nepārtrauktību (Gaikwad, 2024) (Shaikh, 2025).
   3. Izmantojot rādītājus kā sistēmas pieejamības laiku un kļūdu biežumu.
4. Drošība
   1. Drošība ir būtiska, lai aizsargātu sensitīvus datus no neautorizētas piekļuves.
   2. Drošības mehānismi, piemēram, piekļuves kontrole un šifrēšana, ir nepieciešami, lai nodrošinātu datu konfidencialitāti un integritāti (Besa, 2024).
   3. Novērtējot autentifikācijas un autorizācijas mehānismus, kā arī šifrēšanas līmeņus.
5. Lietojamība
   1. Darbiniekam draudzīga saskarne veicina efektīvāku darbu ar sistēmu.
   2. Lietojamība ietekmē darbinieka spēju izmantot sistēmu un sistēmas pieņemšanu organizācijā (Hariri & Norouzi, 2011).
   3. Izmantojot lietotāju aptaujas un uzdevumu izpildes laika mērījumus.
6. Atbalsts un dokumentācija
   1. Datubāzes izstrādātāju atbalsts un dokumentācija atvieglo sistēmas ieviešanu un uzturēšanu.
   2. Pieejams tehniskais atbalsts un detalizēta dokumentācija ir svarīgi faktori sistēmas efektīvai izmantošanai (Besa, 2024).
   3. Novērtējot dokumentācijas pilnīgumu un kvalitāti, kā arī atbalsta pieejamību.
7. Integrācija
   1. Spēja integrēties ar citām sistēmām ir būtisks faktors mūsdienu IT infrastruktūrā.
   2. Integrācijas spējas ļauj DBPS darboties kopā ar citām lietojumprogrammām un pakalpojumiem, veicinot datu apmaiņu un procesu automatizāciju (Besa, 2024) (Gaikwad, 2024).
   3. Izvērtējot pieejamos API un atbalstītos datu formātus.
8. Kopiena
   1. Aktīva kopiena veicina problēmu ātru risināšanu un zināšanu apmaiņu.
   2. Plaša lietotāju bāze nodrošina pieejamu palīdzību un resursus sistēmas izmantošanai un attīstībai (Besa, 2024).
   3. Novērtējot pieejamo resursu, forumu daudzumu un to aktivitāti.
9. Izmaksas
   1. Jāizvērtē izmaksas, kuras varētu būt vienreizēja maksa vai abonaments.
   2. Kopējās izmaksas ietver licencēšanas, aparatūras, apmācību un uzturēšanas izdevumus, kas ietekmē sistēmas rentabilitāti (Besa, 2024).
   3. Aprēķinot visas saistītās izmaksas noteiktā laika periodā.
10. Datu konsistence un integritāte
    1. Nodrošina datu precizitāti un uzticamību.
    2. Datu integritātes mehānismi, piemēram, ierobežojumi un transakciju pārvaldība (Gaikwad, 2024).
    3. Izmantojot datu kvalitātes rādītājus un validācijas pārbaudes.
11. Transakciju pārvaldība
    1. Efektīva transakciju pārvaldība nodrošina datu integritāti un sistēmas uzticamību.
    2. ACID īpašības ir būtiskas drošai datu apstrādei (Taipalus, 2023) (UnityOne.ai, 2024).
    3. Novērtējot transakciju apstrādes laiku un kļūdu biežumu.
12. Datu dublēšana
    1. Svarīgi nodrošināt datu aizsardzību pret zaudējumiem un sistēmas kļūmēm.
    2. Efektīvi rezerves kopēšanas un atkopšanas mehānismi samazina sistēmas nepieejamības laiku un datu zuduma risku (Shaikh, 2025).
    3. Izmantojot atkopšanas laika mērījumus un rezerves kopēšanas regularitātes pārbaudes.
13. Atbilstība standartiem un regulām
    1. Atbilstība nozares standartiem un regulām nodrošina minimālās noteiktās juridiskās prasības.
    2. Sistēmas, kas atbilst tādiem standartiem kā GDPR vai HIPAA, nodrošina datu aizsardzību un atbilstību likumdošanai (Besa, 2024).
    3. Veicot atbilstības pārbaudes.
14. Datu migrācija
    1. Spēja viegli migrēt datus starp sistēmām ir būtisks faktors modernizācijas un integrācijas procesos.
    2. Efektīvas datu migrācijas iespējas samazina pārejas riskus un nodrošina datu nepārtrauktību (Taipalus, 2023) (Catalog, 2025).
    3. Novērtējot migrācijas rīku pieejamību.
15. Datu indeksēšana
    1. Tieši ietekmē datu izgūšanas ātrumu, optimizējot vaicājumu izpildi un resursu izmantošanu.
    2. Indeksēšanas stratēģijām (B-tree, Hash, Bitmap u.c.) ir būtiska nozīme vaicājumu izpildes veiktspējā, īpaši pie liela datu apjoma un kompleksiem vaicājumiem (Taipalus, 2023).
    3. Analizējot pieejamo indeksu tipu daudzveidību, indeksēšanas konfigurācijas iespējas un vaicājumu izpildes ātrumu pirms un pēc indeksu izmantošanas, ar metodēm kā EXPLAIN vaicājumu analīze, I/O operāciju skaits, indeksa atjaunošanas laiks.

**1. tabula**

**Kritēriju tabula**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Kritērijs** | **Svars** | **3 punkti** | **2 punkti** | **1 punkts** |
| 1. Veiktspēja | 8 | Ļoti augsta | Vidēja | Zema |
| 2. Mērogojamība | 5 | Horizontāli un vertikāli | Daļēji mērogojama | Nevar vai grūti |
| 3. Uzticamība | 4 | Mazs nepieejamības laiks | Biežāk nepieejama | Neuzticama |
| 4. Drošība | 8 | Pieejami daudz un dažādi risinājumi | Daži risinājumi | Slikta |
| 5. Lietojamība | 10 | Intuitīva saskarne un labs UX | Vidēji saprotama | Sarežģīta |
| 6. Atbalsts un dokumentācija | 10 | Plaša dokumentācija un aktīvs izstrādātāju atbalsts | Ierobežots | Nav pieejams vai slikts |
| 7. Integrācija | 5 | Atbalsta API un ETL rīkus | Ierobežots | Nav iespēju |
| 8. Kopiena | 10 | Liela un aktīva kopiena un daudz resursi | Vidēja kopiena un resursi | Neesoša vai neaktīva |
| 9. Izmaksas | 12 | Zemas | Vidējas | Ļoti augstas |
| 10. Datu konsistence un integritāte | 5 | ACID pilna ievērošana | Daļēja ievērošana | Nav ievērošanas |
| 11. Transakciju pārvaldība | 4 | Komplekss transakciju atbalsts | Vienkāršs | Nav atbalsts |
| 12. Datu dublēšana | 4 | Automātiska, dinamiska | Manuāla | Nav |
| 13. Atbilstība standartiem un regulām | 7 | Pilnībā atbilst | Daļēji atbilst | Nav atbilstības |
| 14. Datu migrācija | 4 | Plašs atbalsts migrācijai starp sistēmām | Ierobežots atbalsts | Slikta vai nav iespējama |
| 15. Datu indeksēšana | 4 | Daudzveidīgi, pielāgojami indeksi | Daļēji konfigurējami | Ierobežoti vai nav indeksu |

## 2.2. Datubāzu pārvaldības sistēmas

### 2.2.1. Oracle Database

Oracle Database ir viena no visplašāk izmantotajām DBPS uzņēmumos. To izstrādā un uztura Oracle Corporation. Tā tika izlaista 1979. gadā. Oracle Database ir daudzmodeļu DBPS, kas atbalsta gan relāciju, gan objektu - relāciju datu modeļus. Tā ir pazīstama ar augstu veiktspēju, drošību un mērogojamību, padarot to piemērotu lieliem uzņēmumiem. Oracle Database 23c versija piedāvā uzlabotu JSON atbalstu, blokķēdes drošību un AI vadītu automatizāciju. Oracle ir līderis DBPS sistēmās vairākas desmitgades, un tās jaunākās versijas turpina ieviest inovācijas datu pārvaldībā (Wikipedia, 2025).

**2. tabula**

**Oracle Database izvērtējums**

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1. Veiktspēja | 3 punkti | autora subjektīvs viedoklis |
| 2. Mērogojamība | 3 punkti | autora subjektīvs viedoklis |
| 3. Uzticamība | 3 punkti | autora subjektīvs viedoklis |
| 4. Drošība | 3 punkti | autora subjektīvs viedoklis |
| 5. Lietojamība | 2 punkti | autora subjektīvs viedoklis |
| 6. Atbalsts un dokumentācija | 3 punkti | autora subjektīvs viedoklis |
| 7. Integrācija | 3 punkti | autora subjektīvs viedoklis |
| 8. Kopiena | 2 punkti | autora subjektīvs viedoklis |
| 9. Izmaksas | 1 punkts | autora subjektīvs viedoklis |
| 10. Datu konsistence un integritāte | 3 punkti | autora subjektīvs viedoklis |
| 11. Transakciju pārvaldība | 3 punkti | autora subjektīvs viedoklis |
| 12. Datu dublēšana | 3 punkti | autora subjektīvs viedoklis |
| 13. Atbilstība standartiem un regulām | 3 punkti | autora subjektīvs viedoklis |
| 14. Datu migrācija | 2 punkti | autora subjektīvs viedoklis |
| 15. Datu indeksēšana | 3 punkti | autora subjektīvs viedoklis |

### 2.2.2. MySQL

MySQL izstrādāja MySQL AB, bet tagad pieder Oracle Corporation. Tika izstrādāts 1995. gadā. MySQL ir atvērtā koda relāciju DBPS, kas ir plaši izmantota tīmekļa lietojumprogrammās. Tā ir pazīstama ar vienkāršu uzstādīšanu, lietojamību un plašu kopienas atbalstu. MySQL 9.0 versija piedāvā paralēlu vaicājumu izpildi un uzlabotu replikāciju. MySQL ir populāra izvēle mazām un vidējām uzņēmumu lietojumprogrammām, dēļ tās efektivitātes un zemajām izmaksām (Mcare Technologie, 2024).

**3. tabula**

**MySQL izvērtējums**

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1. Veiktspēja | 2 punkti | autora subjektīvs viedoklis |
| 2. Mērogojamība | 2 punkti | autora subjektīvs viedoklis |
| 3. Uzticamība | 2 punkti | autora subjektīvs viedoklis |
| 4. Drošība | 2 punkti | autora subjektīvs viedoklis |
| 5. Lietojamība | 3 punkti | autora subjektīvs viedoklis |
| 6. Atbalsts un dokumentācija | 3 punkti | autora subjektīvs viedoklis |
| 7. Integrācija | 3 punkti | autora subjektīvs viedoklis |
| 8. Kopiena | 3 punkti | autora subjektīvs viedoklis |
| 9. Izmaksas | 3 punkti | autora subjektīvs viedoklis |
| 10. Datu konsistence un integritāte | 2 punkti | autora subjektīvs viedoklis |
| 11. Transakciju pārvaldība | 2 punkti | autora subjektīvs viedoklis |
| 12. Datu dublēšana | 2 punkti | autora subjektīvs viedoklis |
| 13. Atbilstība standartiem un regulām | 2 punkti | autora subjektīvs viedoklis |
| 14. Datu migrācija | 2 punkti | autora subjektīvs viedoklis |
| 15. Datu indeksēšana | 2 punkti | autora subjektīvs viedoklis |

### 2.2.3. Microsoft SQL Server

Microsoft SQL Server izstrādāja Microsoft Corporation 1989. gadā. Microsoft SQL Server ir relāciju DBPS, kas integrējas ar Microsoft ekosistēmu, piedāvājot augstu drošību un veiktspēju. SQL Server 2025 versija ietver AI atbalstītu vaicājumu optimizāciju un mākoņa integrāciju. SQL Server ir populāra izvēle uzņēmumiem, kas izmanto Microsoft tehnoloģijas, nodrošinot ciešu integrāciju ar citiem Microsoft produktiem (Linhartová, 2025).

**4. tabula**

**Microsoft SQL Server izvērtējums**

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1. Veiktspēja | 3 punkti | autora subjektīvs viedoklis |
| 2. Mērogojamība | 2 punkti | autora subjektīvs viedoklis |
| 3. Uzticamība | 3 punkti | autora subjektīvs viedoklis |
| 4. Drošība | 3 punkti | autora subjektīvs viedoklis |
| 5. Lietojamība | 2 punkti | autora subjektīvs viedoklis |
| 6. Atbalsts un dokumentācija | 3 punkti | autora subjektīvs viedoklis |
| 7. Integrācija | 3 punkti | autora subjektīvs viedoklis |
| 8. Kopiena | 2 punkti | autora subjektīvs viedoklis |
| 9. Izmaksas | 2 punkti | autora subjektīvs viedoklis |
| 10. Datu konsistence un integritāte | 3 punkti | autora subjektīvs viedoklis |
| 11. Transakciju pārvaldība | 3 punkti | autora subjektīvs viedoklis |
| 12. Datu dublēšana | 3 punkti | autora subjektīvs viedoklis |
| 13. Atbilstība standartiem un regulām | 3 punkti | autora subjektīvs viedoklis |
| 14. Datu migrācija | 2 punkti | autora subjektīvs viedoklis |
| 15. Datu indeksēšana | 3 punkti | autora subjektīvs viedoklis |

### 2.2.4. PostgreSQL

PostgreSQL izstrādāja PostgreSQL Global Development Group 1996. gadā. PostgreSQL ir atvērtā koda relāciju DBPS, kas atbalsta pielāgošanu. PostgreSQL piedāvā augstu atbilstību SQL standartiem un atbalsta dažādus datu tipus. PostgreSQL 17 versija ietver vektoru meklēšanu un uzlabotu JSONB indeksēšanu. PostgreSQL ir pazīstama ar savu stabilitāti un dinamiskumu, padarot to par populāru izvēli dažādiem lietojumiem (Wikipedia, 2025).

**5. tabula**

**PostgreSQL izvērtējums**

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1. Veiktspēja | 3 punkti | autora subjektīvs viedoklis |
| 2. Mērogojamība | 2 punkti | autora subjektīvs viedoklis |
| 3. Uzticamība | 3 punkti | autora subjektīvs viedoklis |
| 4. Drošība | 3 punkti | autora subjektīvs viedoklis |
| 5. Lietojamība | 2 punkti | autora subjektīvs viedoklis |
| 6. Atbalsts un dokumentācija | 3 punkti | autora subjektīvs viedoklis |
| 7. Integrācija | 3 punkti | autora subjektīvs viedoklis |
| 8. Kopiena | 3 punkti | autora subjektīvs viedoklis |
| 9. Izmaksas | 3 punkti | autora subjektīvs viedoklis |
| 10. Datu konsistence un integritāte | 3 punkti | autora subjektīvs viedoklis |
| 11. Transakciju pārvaldība | 3 punkti | autora subjektīvs viedoklis |
| 12. Datu dublēšana | 3 punkti | autora subjektīvs viedoklis |
| 13. Atbilstība standartiem un regulām | 3 punkti | autora subjektīvs viedoklis |
| 14. Datu migrācija | 2 punkti | autora subjektīvs viedoklis |
| 15. Datu indeksēšana | 3 punkti | autora subjektīvs viedoklis |

### 2.2.5. MongoDB

MongoDB izstrādāja MongoDB Inc. (sākumā saukts 10gen) 2009. gadā. MongoDB ir dokumentu orientēta NoSQL DBPS, kas piedāvā dinamisku datu modeli un horizontālu mērogojamību. MongoDB atbalsta ad-hoc vaicājumus, replikāciju un ”sharding”. MongoDB 7.2 versija uzlabo “time-series” veiktspēju un drošības funkcijas. MongoDB ir kļuvusi par populāru izvēli lietojumprogrammām, kas prasa vienkāršu un dinamisku datubāzi.

**6. tabula**

**MongoDB izvērtējums**

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1. Veiktspēja | 2 punkti | autora subjektīvs viedoklis |
| 2. Mērogojamība | 3 punkti | autora subjektīvs viedoklis |
| 3. Uzticamība | 2 punkti | autora subjektīvs viedoklis |
| 4. Drošība | 2 punkti | autora subjektīvs viedoklis |
| 5. Lietojamība | 3 punkti | autora subjektīvs viedoklis |
| 6. Atbalsts un dokumentācija | 3 punkti | autora subjektīvs viedoklis |
| 7. Integrācija | 2 punkti | autora subjektīvs viedoklis |
| 8. Kopiena | 3 punkti | autora subjektīvs viedoklis |
| 9. Izmaksas | 3 punkti | autora subjektīvs viedoklis |
| 10. Datu konsistence un integritāte | 2 punkti | autora subjektīvs viedoklis |
| 11. Transakciju pārvaldība | 2 punkti | autora subjektīvs viedoklis |
| 12. Datu dublēšana | 3 punkti | autora subjektīvs viedoklis |
| 13. Atbilstība standartiem un regulām | 2 punkti | autora subjektīvs viedoklis |
| 14. Datu migrācija | 2 punkti | autora subjektīvs viedoklis |
| 15. Datu indeksēšana | 3 punkti | autora subjektīvs viedoklis |

### 2.2.6. IBM Db2

IBM Db2 izstrādāja IBM Corporation 1983. gadā. IBM Db2 ir relāciju DBPS, kas atbalsta arī objektu-relāciju un ne-relāciju datu struktūras. IBM Db2 piedāvā funkcijas kā BLU Acceleration un adaptīvo kompresiju, kas uzlabo analītikas veiktspēju un samazina datu glabāšanas izmaksas. IBM Db2 ir bijusi nozīmīga DBPS uzņēmumu vidē, īpaši finanšu un veselības aprūpes nozarēs (Wikipedia, 2025).

**7. tabula**

**IBM Db2 izvērtējums**

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1. Veiktspēja | 3 punkti | autora subjektīvs viedoklis |
| 2. Mērogojamība | 2 punkti | autora subjektīvs viedoklis |
| 3. Uzticamība | 3 punkti | autora subjektīvs viedoklis |
| 4. Drošība | 3 punkti | autora subjektīvs viedoklis |
| 5. Lietojamība | 2 punkti | autora subjektīvs viedoklis |
| 6. Atbalsts un dokumentācija | 3 punkti | autora subjektīvs viedoklis |
| 7. Integrācija | 3 punkti | autora subjektīvs viedoklis |
| 8. Kopiena | 2 punkti | autora subjektīvs viedoklis |
| 9. Izmaksas | 1 punkts | autora subjektīvs viedoklis |
| 10. Datu konsistence un integritāte | 3 punkti | autora subjektīvs viedoklis |
| 11. Transakciju pārvaldība | 3 punkti | autora subjektīvs viedoklis |
| 12. Datu dublēšana | 3 punkti | autora subjektīvs viedoklis |
| 13. Atbilstība standartiem un regulām | 3 punkti | autora subjektīvs viedoklis |
| 14. Datu migrācija | 2 punkti | autora subjektīvs viedoklis |
| 15. Datu indeksēšana | 3 punkti | autora subjektīvs viedoklis |

### 2.2.7. SQLite

SQLite izstrādāja Dwayne Richard Hipp 2000. gadā. SQLite ir maza, “embedded” DBPS, kas neprasa atsevišķu serveri vai administrēšanu. Tā ir piemērota mobilajām lietojumprogrammām un iebūvētām sistēmām. SQLite izmanto vienu failu datu glabāšanai un atbalsta ACID transakcijas. SQLite ir plaši izmantota dažādās lietojumprogrammās, pateicoties tās vienkāršībai un uzticamībai (Wikipedia, 2025).

**8. tabula**

**SQLite izvērtējums**

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1. Veiktspēja | 2 punkti | autora subjektīvs viedoklis |
| 2. Mērogojamība | 1 punkts | autora subjektīvs viedoklis |
| 3. Uzticamība | 2 punkti | autora subjektīvs viedoklis |
| 4. Drošība | 2 punkti | autora subjektīvs viedoklis |
| 5. Lietojamība | 3 punkti | autora subjektīvs viedoklis |
| 6. Atbalsts un dokumentācija | 2 punkti | autora subjektīvs viedoklis |
| 7. Integrācija | 1 punkts | autora subjektīvs viedoklis |
| 8. Kopiena | 2 punkti | autora subjektīvs viedoklis |
| 9. Izmaksas | 3 punkti | autora subjektīvs viedoklis |
| 10. Datu konsistence un integritāte | 3 punkti | autora subjektīvs viedoklis |
| 11. Transakciju pārvaldība | 2 punkti | autora subjektīvs viedoklis |
| 12. Datu dublēšana | 1 punkts | autora subjektīvs viedoklis |
| 13. Atbilstība standartiem un regulām | 2 punkti | autora subjektīvs viedoklis |
| 14. Datu migrācija | 2 punkti | autora subjektīvs viedoklis |
| 15. Datu indeksēšana | 2 punkti | autora subjektīvs viedoklis |

## 2.3. Datubāzu pārvaldības sistēmu salīdzinošā analīze

**9. tabula**

**DBPS salīdzinājums**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DBPS**  **Kritērijs** | **Svars** | **Oracle Database** | **MySQL** | **Microsoft SQL Server** | **PostgreSQL** | **MongoDB** | **IBM Db2** | **SQLite** |
| 1. Veiktspēja | 8 | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 2 punkti | 3 punkti | 2 punkti |
| 2. Mērogojamība | 5 | 3 punkti | 2 punkti | 2 punkti | 2 punkti | 3 punkti | 2 punkti | 1 punkts |
| 3. Uzticamība | 4 | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 2 punkti | 3 punkti | 2 punkti |
| 4. Drošība | 8 | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 2 punkti | 3 punkti | 2 punkti |
| 5. Lietojamība | 10 | 2 punkti | 3 punkti | 2 punkti | 2 punkti | 3 punkti | 2 punkti | 3 punkti |
| 6. Atbalsts un dokumentācija | 10 | 3 punkti | 3 punkti | 3 punkti | 3 punkti | 3 punkti | 3 punkti | 2 punkti |
| 7. Integrācija | 5 | 3 punkti | 3 punkti | 3 punkti | 3 punkti | 2 punkti | 3 punkti | 1 punkts |
| 8. Kopiena | 10 | 2 punkti | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 2 punkti | 2 punkti |
| 9. Izmaksas | 12 | 1 punkts | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 1 punkts | 3 punkti |
| 10. Datu konsistence un integritāte | 5 | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 2 punkti | 3 punkti | 3 punkti |
| 11. Transakciju pārvaldība | 4 | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 2 punkti | 3 punkti | 2 punkti |
| 12. Datu dublēšana | 4 | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 3 punkti | 3 punkti | 1 punkts |
| 13. Atbilstība standartiem un regulām | 7 | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 2 punkti | 3 punkti | 2 punkti |
| 14. Datu migrācija | 4 | 2 punkti | 2 punkti | 2 punkti | 2 punkti | 2 punkti | 2 punkti | 2 punkti |
| 15. Datu indeksēšana | 4 | 3 punkti | 2 punkti | 3 punkti | 3 punkti | 3 punkti | 3 punkti | 2 punkti |
| Vērtējums | 100 | 252 (40) | 247 (35) | 259 (40) | 281 (42) | 255 (37) | 247 (39) | 213 (30) |

9. tabulas rindiņā „Svars” ierakstīti katra kritērija būtiskuma skaitlisks vērtējums, atbilstoši Code Moodle projekta prasībām. Svara koeficientu vērtējumi iegūti projekta komandas savstarpējā diskusijā, un tie atspoguļo visu izstrādātāju komandas viedokli. Kā redzams pēc analīzes tabulas datiem, Code Moodle projekta izstrādei pēc gala punktiem vispiemērotākā DBPS ir PostgreSQL, kas vērtējumā ir ieguvis augstāko rezultātu gan ar svaru un bez svaru (bez svaru aprēķins ir redzams iekavās) aprēķinā – 281 (42).

## 2.4. Analīzes rezultāti

Pēc detalizētas septiņu populārāko datubāzu pārvaldības sistēmu analīzes, var izdarīt vairākus secinājumus no konkrētām lietojuma vajadzībām.

Pirmkārt, Oracle Database izceļas ar veiktspēju, uzticamību, drošību un plašu funkcionalitāti, padarot to par ideālu risinājumu lieliem uzņēmumiem un aplikācijām. Tomēr šī sistēma nav paredzēta maziem un ātriem risinājumiem.

PostgreSQL un Microsoft SQL Server piedāvā līdzīgu funkcionalitāti, taču PostgreSQL priekšrocība ir tās atvērtā koda princips. SQL Server, savukārt, ir piemērots tiem, kas jau izmanto un vēlas Microsoft ekosistēmas produktus.

MySQL un SQLite ir labi vienkāršākām sistēmām. MySQL ir piemērots tīmekļa lietotnēm, kur nav nepieciešama augsta transakciju kontrole. SQLite ir lielisks risinājums mobilajām sistēmām, kur nepieciešama minimāla konfigurācija un servera infrastruktūra.

MongoDB stiprās puses ir darbā ar mazstrukturētiem datiem. Tā ir īpaši ieteicama sistēmām ar reāllaika datu apstrādes vajadzībām.

IBM Db2, lai gan spēcīga pēc veiktspējas un drošības, var būt sarežģīta ieviešanai. Tomēr tā ir laba izvēle uzņēmumiem ar nopietnām datu integritātes un atbilstības prasībām, piemēram, bankām vai veselības aprūpes iestādēm.

Izvēloties DBPS, nav noteiktais labākais risinājums, tas ir atkarīgs no konkrētā lietojuma, datu apjoma, drošības prasībām, budžeta un tehniskās vides. Tāpēc ņemot vērā, mūsu gadījumā divas galvenās lietas – komandas pieredzi un izmaksas – mums visspiemērotākā DBPS ir MySQL. Lai gan mums arī ir pieredze izmantojot MongoDB, tomēr šim projektam ir nepieciešama relāciju datubāze.

# Secinājumi

Studiju projekta pētījumā izvirzītais mērķis ir sasniegts, un priekš satura pāvaldības sistēmas Code Moodle izstrādei, piemērotāko datubāzu pārvaldības sistēmu ir izdevies noteikt. Salīdzinot un izpētot ap divdesmit datubāzu pārvaldības sistēmas, par piemērotāko ir noteikta MySQL.

Pētījuma gaitā ir apzinātas apmērām divdesmit datubāzu pārvaldības sistēmas. Informācijas iegūšanai tika izmantoti ap divdesmit informācijas avoti. Jāsecina, ka datubāzes izstrādes problēmas ir ļoti aktuālas un pieejamā informācija par šiem jautājumiem ir ļoti plaša.

Pamatojoties uz literatūras un informācijas avotu analīzes rezultātiem, darbā ir noteikti piecpadsmit datubāzu pārvaldības sistēmu novērtēšanas kritēriji. Deviņi no tiem ir izmantojami jebkura programmatūras izstrādes rīka novērtēšani, pārējie – specifiski datubāzu sistēmu izstrādei.

Pētījuma gaitā izveidoto salīdzināšānas analīzes tabulu var izmantot arī citos datubāzes projektos, savu piemērotāko datubāzes pārvaldības sistēmas izvēlei (izstrādātājiem tikai jāieraksta savi svara koeficienti).

Šī darba teorētisko pamatojumu var izmantot kā mācību materiālu tiem, kuri apgūst datubāzu pamatus un vēlas uzzināt vairāk par Code Moodle projekta vajadzībām un veidošanu. Šo darbu varētu papildināt ar citiem Code Moodle projekta vajadzīgajiem rīkiem, piemēram, kā programmēšanas valodu analīzi, kur varētu izmantot iepriekš definēto kritēriju tabulu, specifiski deviņus definētos vispārīgos punktus.

# Jēdzienu skaidrojumi un saīsinājumi

* Datubāze (DB) – savstarpēji saistītu informacionālu objektu tematisks kopums, kas ar speciālas pārvaldības sistēmas starpniecību izveidots un organizēts tā, lai nodrošinātu tajā ievadītās informācijas izguvi, veiktu tās atlasi un kārtošanu.
* Datubāzu pārvaldības sistēma (DBPS) – programmatūras rīks, kas nodrošina iespēju veidot, uzturēt un pārvaldīt datubāzes. DBPS ļauj lietotājiem un lietojumprogrammām piekļūt datiem drošā, konsekventā un efektīvā veidā. DBPS ietver funkcijas datu glabāšanai, transakciju apstrādei, piekļuves kontrolei, atjaunošanai un datu vaicājumu valodai.
* Relāciju datubāze (RDB) – datubāzes veids, kurā dati tiek organizēti tabulās (relācijās), un katra tabula sastāv no rindu un kolonnu kopuma. Tabulas var būt savstarpēji saistītas ar atslēgām (piemēram, primārā un ārējā atslēga). Šāda tipa DBPS izmanto SQL vaicājumu valodu.
* NoSQL – “Not Only SQL” jeb ne tikai SQL ir datubāzu grupa, kas nav balstīta uz relāciju modeli. NoSQL datubāzes ir piemērotas darbam ar liela apjoma, dažādas struktūras vai mainīgiem datiem. Tajās tiek izmantoti dokumentu, atslēga-vērtība, kolonnu vai grafu datu modeļi.
* SQL (Structured Query Language) – strukturēta vaicājumu valoda, ko izmanto relāciju datubāzēs datu izgūšanai, modificēšanai, pievienošanai un dzēšanai. SQL ir industrijas standarts, kas atbalstīts lielākajā daļā relāciju DBPS.
* ACID īpašības – četras pamatprasības datu transakcijām: atomiskums (atomicity), konsekvence (consistency), izolācija (isolation), izturība (durability). Šīs īpašības nodrošina datu integritāti un uzticamību.
* Mērogojamība (Scalability) – DBPS spēja pielāgoties pieaugošam lietotāju skaitam vai datu apjomam, nezaudējot veiktspēju. To iedala vertikālā (resursu pievienošana vienai iekārtai) un horizontālā (pievienojot vairākas sistēmas) mērogojamībā.
* Rezerves kopēšana (Backup) – process, kura laikā tiek veidota datu kopija, lai nodrošinātu iespēju tās atjaunot sistēmas bojājuma vai datu zuduma gadījumā.
* Replikācija (Replication) – datu kopēšanas process uz vairākām lokācijām vai serveriem, lai nodrošinātu datu pieejamību, drošību un slodzes līdzsvarošanu.
* ETL (Extract, Transform, Load) – datu integrācijas process, kurā dati tiek iegūti no avotiem (Extract), pārveidoti piemērotā formātā (Transform) un ielādēti mērķa sistēmā vai datu noliktavā (Load).
* DBaaS (Database as a Service) – mākoņpakalpojums, kas nodrošina DBPS funkcionalitāti kā servisu, neprasot lietotājam administrēt aparatūru vai uzstādīt programmatūru.
* API (Application Programming Interface) – lietojumprogrammas saskarne, tas ir noteikumu, protokolu un rīku kopums, kas nosaka, kā programmatūras komponentēm savstarpēji jāsazinās. API ļauj vienai programmai piekļūt citas programmas funkcionalitātei vai datiem, piemēram, DBPS var nodrošināt API, lai trešo pušu lietotnes varētu veikt datu vaicājumus, ievadi vai atjaunināšanu.

# Izmantotā literatūra

1. Besa, H. (2024. gada 18. Septembris). *12 Criteria for Evaluating Database Management Software*. Ielādēts no Softlist.io: https://www.softlist.io/evaluating-database-management-software
2. Catalog. (2025. gada 6. Marts). *Data Quality Metrics: 5 Metrics to Monitor the Health of Your Data Estate*. Ielādēts no Catalog: https://www.castordoc.com/data-strategy/data-quality-metrics-5-metrics-to-monitor-the-health-of-your-data-estate
3. CelerData. (2024. gada 4. Novembris). *How Database Management Systems Have Evolved Over Time*. Ielādēts no CelerData: https://celerdata.com/glossary/how-database-management-systems-have-evolved-over-time
4. Gaikwad, C. (2024. gada 3. Septembris). *How to Evaluate Drug Safety Database Performance with Key Metrics*. Ielādēts no Datacreds: https://www.datacreds.com/post/how-to-evaluate-drug-safety-database-performance-with-key-metrics
5. Hariri, N., & Norouzi, Y. (2011). Determining evaluation criteria for digital libraries' user interface: a review. *The Electronic Library*, 698–722.
6. Linhartová, B. (2025. gada 20. Marts). *The 10 best databases of 2025: features and latest changes*. Ielādēts no Baremon: https://www.baremon.eu/10-best-databases-of-2025/
7. Mcare Technologie. (2024. gada 8. Oktobris). *Top Database Management Software in 2025*. Ielādēts no Mcare Technologie: https://mcaretechnologie.com/top-database-management-software-in-2025/
8. Perdue, T. (2024. gada 18. Aprīlis). *An Overview of NoSQL Databases*. Ielādēts no Lifewire: https://www.lifewire.com/nosql-an-overview-of-nosql-databases-2495393
9. Quickbase. (2024). *A Timeline of Database History & Database Management*. Ielādēts no https://www.quickbase.com/articles/timeline-of-database-history
10. Robb, D. (2023. gada 11. Jūlijs). *Top 5 Current Database Trends*. Ielādēts no Datamation: https://www.datamation.com/cloud/current-database-trends/
11. Sande, F. (2025). *Network Database, Relational DB, and Graph DB Compared*. Ielādēts no Raima: https://raima.com/network-database-relational-db-and-graph-db-compared/
12. Shaikh, F. (2025. gada 14. Aprīlis). *Database Monitoring Metrics: What to Track & Why It Matters*. Ielādēts no Last9: https://last9.io/blog/database-monitoring-metrics
13. Taipalus, T. (2023). Database management system performance comparisons: A systematic literature review. *Journal of Systems and Software*, 36.
14. Team EMB. (2024. gada 23. Jūlijs). *Emerging Trends in Database Management Systems*. Ielādēts no EMB Global: https://blog.emb.global/emerging-trends-in-database-management-systems/
15. UnityOne.ai. (2024. gada 19. Novembris). *A Guide To Database Monitoring - Top 5 Metrics To Consider*. Ielādēts no UnityOne.ai: https://www.unityone.ai/a-guide-to-database-monitoring-top-5-metrics-to-consider
16. Wikipedia. (2025. gada 21. Maijs). *IBM Db2*. Ielādēts no Wikipedia: https://en.wikipedia.org/wiki/IBM\_Db2
17. Wikipedia. (2025. gada 13. Februāris). *Integrated Data Store*. Ielādēts no Wikipedia: https://en.wikipedia.org/wiki/Integrated\_Data\_Store
18. Wikipedia. (2025. gada 4. Aprīlis). *Oracle Database*. Ielādēts no Wikipedia: https://en.wikipedia.org/wiki/Oracle\_Database
19. Wikipedia. (2025. gada 8. Maijs). *PostgreSQL*. Ielādēts no Wikipedia: https://en.wikipedia.org/wiki/PostgreSQL
20. Wikipedia. (2025. gada 21. Maijs). *SQLite*. Ielādēts no Wikipedia: https://en.wikipedia.org/wiki/SQLite

# II daļa Programmatūras projekts, komandas veikums

# 1. “Code Moodle” projekta apraksts

## 1.1 Projekta mērķis un uzdevumi

Projekta mērķis ir izveidot sistēmu, kura atvieglo studentiem un skolotājiem kodēšanas uzdevumu uzdošanu, veikšanu un vērtēšanu. Projekta uzdevumi ir izveidot datubāzi, datubāzes saskarni un izveidot klienta pusi. Datubāzei ir jāspēj glabāt visus nepieciešamos datus, no lietotāju ielogošanās datiem, līdz kodēšanas uzdevumu iesniegumiem. Datubāzes saskarne ir nepieciešama lai lietotājiem nebūtu tieša pieeja datubāzes savienojumam, procesi varētu tikt automatizēti, piemēram, reģistrācija, klienta pusē pārbauda vai ievaddati ir derīgi, tad datubāzes saskarne pārbauda vai lietotājs ar doto epastu jau nepastāv un ja nē tad izveido lietotāju. Klienta puse ir saskarne kur lietotāji spēs apmeklēt un lietot, klienta puse veic datu vizualizāciju balstoties uz lietotāja lomām, piemēram, skolotāji spēj veidot uzdevumus un moduļus kur skolēni redz tikai pieejamos moduļus, uzdevumus un iesniegumu vēsturi. Klienta puse arī veic datu validāciju, tas ir, vai dati, tādi kā epasts vai parole ir derīgi. Datu validācija neietver datu duplikācijas pārbaudes, to veic datubāzes saskarne.

## Projekta organizatoriskā struktūra

Komandas sastāvs:

* Kārlis Lācītis
* Endijs Dārznieks

**10. tabula**

**Komandas lomu sadalījums, uzdevumi un prasības**

|  |  |  |
| --- | --- | --- |
| Lomas nosaukums | **Dalībnieki** | **Lomas darba uzdevumi** |
| Projekta vadītājs | Kārlis Lācītis | Darba organizācija |
| Sistēmas saskarnes izstrādātājs | Kārlis Lācītis, Endijs Dārznieks | Sistēmas klienta puses izstrāde |
| Sistēmas servera izstrādātājs | Kārlis Lācītis, Endijs Dārznieks | Sistēmas API izstrāde |
| Dizaineris | Endijs Dārznieks | Sistēmas klienta puses izstrāde |
| Analists | Endijs Dārznieks | Prasību noteikšana |
| Datubāzes modelētājs | Kārlis Lācītis | Datubāzes modelēšana |
| Testētājs | Kārlis Lācītis, Endijs Dārznieks | Testēšanas žurnāla sagatave, LRG izstrāde |
| Arhitekts | Endijs Dārznieks | Projekta publiskās hostēšanas nodrošināšana |

Komandas komunikācijas rīki:

* Sarunas klātienē
* Discord
* Trello

## Projekta nodevumi

Projekta nodevumi:

1. PPS
2. PPA
3. PPPP
4. Intervijas protokols
5. Testēšanas žurnāls
6. LRG
7. Individuālais pētījums / darba apraksts.
8. Klienta puses pirmkods
9. Servera puses pirmkods

Visi nodevumi ir pieejami <https://github.com/EndijsD/code-moodle>

# 2. Projekta nodevumu apraksts

Nodevumi iedalāmi trīs grupās(kopīgs):

* Programmatūras projekta pārvaldības plāns (PPPP);
* programmatūras dokumentācija:
  + Programmatūras prasību specifikācija (PPS), Programmatūras projektējuma apraksts (PPA), lietotāja rokasgrāmata (LRG),
  + testēšanas dokumentācija (TST) (testpiemēri, testēšanas žurnāls, Problēmziņojumi un Problēmu/kļūdu ziņojumu reģistrācijas žurnāls, testēšanas kopsavilkums);
* pirmkodi – Pirmkods tiek iedalīts divās daļās :
  + Klienta puses pirmkods – satur pirmkodu mājaslapas struktūrai, datu vizualizācijai un vaicājumu izsaukšanai un projekta servera pusi.
  + Servera puses pirmkods – pirmkods apstrādā vaicājumus no klienta puses, veic pārbaudes un veic datu manipulācijas datubāzē

Datubāze un tās shēma – nodevumā nāks MySQL datubāzes shēmas fails, kur ir iespējams apskatīt datubāzes ER shēmu. Ar rīku forward engineering var automatizēt vaicājumu izvedi datubāzes izveidei. Kopā ar shēmu nāk kopā .sql fails kurš satur datubāzes izveidi kopā ar sākumierakstiem testēšanai.

# 3. Autora un pārējo komandas dalībnieku ieguldījums projekta tapšanā

Komandas dalībnieki kopīgi veica sistēmas izstrādi. Endijs Dārznieks veica pamāta jaunu funkcionalitāšu ieviešanu, kur Kārlis Lācītis veica funkciju pielāgošanu eksistējošajai sistēmai. Pie dokumentācijas vairāk laika pavadīja Kārlis Lācītis un pie datubāzes pārstrādes vairāk laika pavadīja Endijs Dārznieks.

Veicu datubāzes pielāgošanu jaunajam sistēmas servisa modelim (SaaS), kura rezultātā man nācās refaktorēt reģistrācijas lapu. Pats izveidoju sistēmas autentifikācijas un failu loģiku. Veicu sistēmas hostēšanu un tās pielāgošanas priekš hostēšanas.

**11. tabula**

**Kopējais procentuālais ieguldījus komandas darbā**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dalībnieks** | **Programmatūra** | | | | | | **Pārvaldība** | **Kopā** |
| **PPS** | **PPA** | **LRG** | **TST** | **Kods** | **DB** | **PPPP** |
| Kārlis Lācīts | 50% | 60% | 80% | 40% | 50% | 20% | 70% | **370%** |
| Endijs Dārznieks | 50% | 40% | 20% | 60% | 50% | 80% | 30% | **330%** |
|  | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **700%** |

**Darba vērtējums**

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi un manā darbā nav pārkāpts likums par autortiesībām.

Darba autors: Endijs Dārznieks ........................ 06.06.2025.

Studiju darbs aizstāvēts RTU Liepājas akadēmija

Dabas un inženierzinātņu centrs

studiju darbu aizstāvēšanas komisijas sēdē ........................................................

(datums)

Novērtējums ...................................................

studiju darba vadītājs: Dzintars Tomsons/Anita Jansone .........................................